The djatoka JPEG 2000 image server

Ryan Chute

Digital Library Research & Prototyping Team
Research Library
Los Alamos National Laboratory

rchute@lanl.gov

Who am I?

- Researcher and software engineer on the Digital Library Research and Prototyping Team at the Research Library of the Los Alamos National Laboratory.
- Research focuses on leveraging existing standards and technologies to develop highly scalable, component-based systems.
- Project manager and developer for aDORe.
- Senior engineer for the Mesur project.
- Over 10 years experience with high resolution digital imaging in the cultural heritage community; from capture, correction, storage to delivery.
- 5 years experience with JPEG 2000 and Kakadu Software
- Previously Technical Project Manager and Systems Engineer for Luna Imaging's Insight Software (1997-2005)

Outline

Presentation breaks down into two parts:

1. aDORe djatoka Project Update
    - Features
    - Adoption
    - Current and Future Development
2. JPEG 2000: Barriers to Adoption
    - What are the perceived issues?
    - Who is currently using JPEG 2000?
    - How are they using the format?
    - How do we encourage adoption?

Part 1 - aDORe djatoka Project Update

Context: The aDORe Project

- Concrete need to design and implement a solution to ingest, store, access the vast and growing collection of the LANL Research Library.
  - Scale, scale, scale!
- Interest in repository interoperability (OpenURL, OAI-PMH)
- Leverage existing standards and technologies to make development and migration more straightforward.
- Use a distributed, component based approach to meet challenges of scale.
- Use Digital Objects, Datastreams, and Surrogate abstractions to characterize content.
- Facilitate a uniform manner for client applications to discover and access content objects available in a group of distributed repositories.
- Provide single repository behavior for a group of distributed repositories.

What is aDORe djatoka?

- Open-source JPEG 2000 image server and dissemination framework
  - Provides Web Service & Java Application Interfaces
- Leverages existing standards and technologies
  - Standards: ISO JPEG 2000 / NISO OpenURL
  - APIs: ImageJ, JAI, OOM
- Provides of an implementation agnostic (e.g. Kakadu, Aware, etc) framework for JPEG 2000 compression and extraction.
- Geared towards reuse through URI-addressability of all image disseminations including regions, rotations, and format transformations
- Provides an extensible service framework for image disseminations

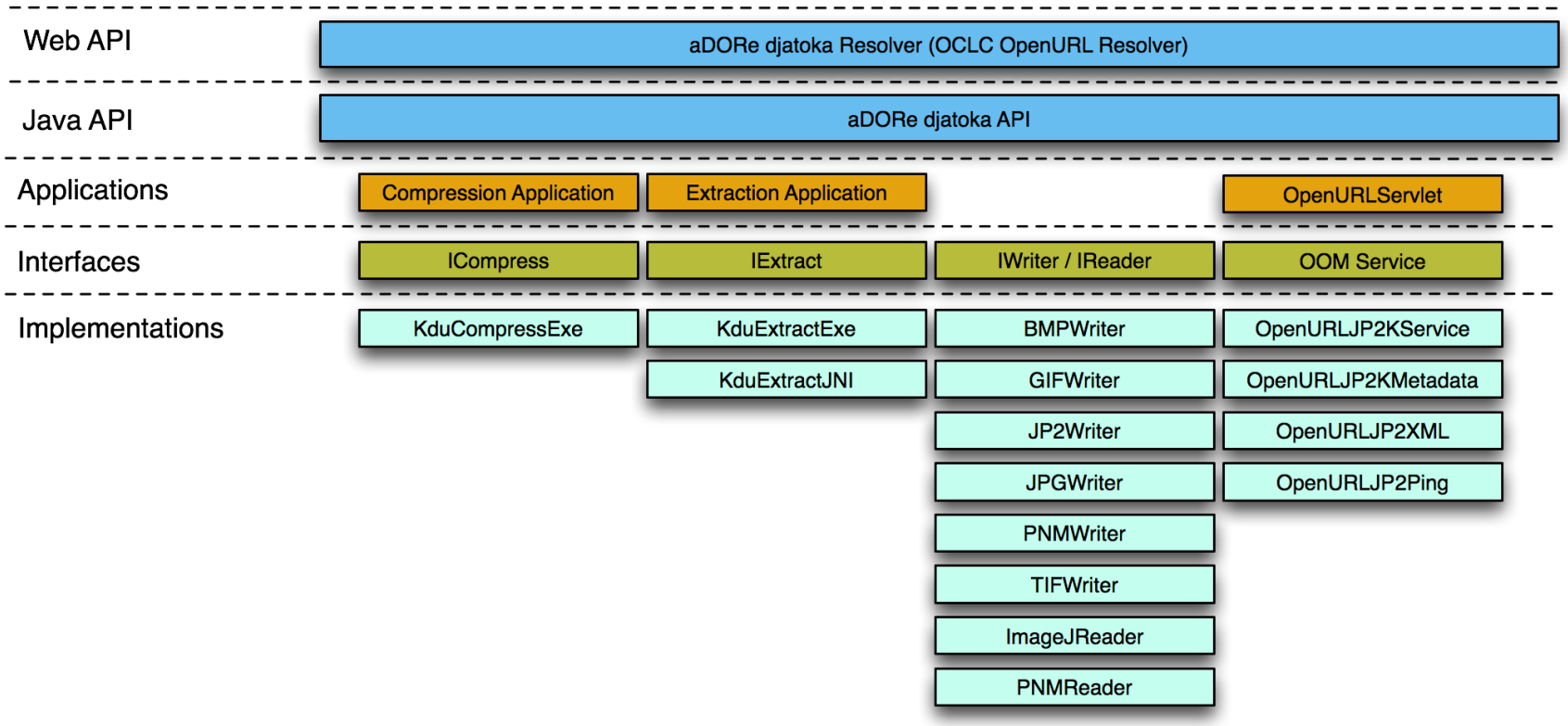
Why aDORe djatoka?

- Lack of open source image server implementations.
- Lack of an easily extensible image dissemination service framework.
- Lack of standard syntax for the URI-addressability of image disseminations including regions, rotations, and format transformations.
- Desire to encourage the adoption of JPEG 2000 as a service and/or archival image file format.
- Desire to develop a community defined open source image dissemination server platform.

Why JPEG 2000?

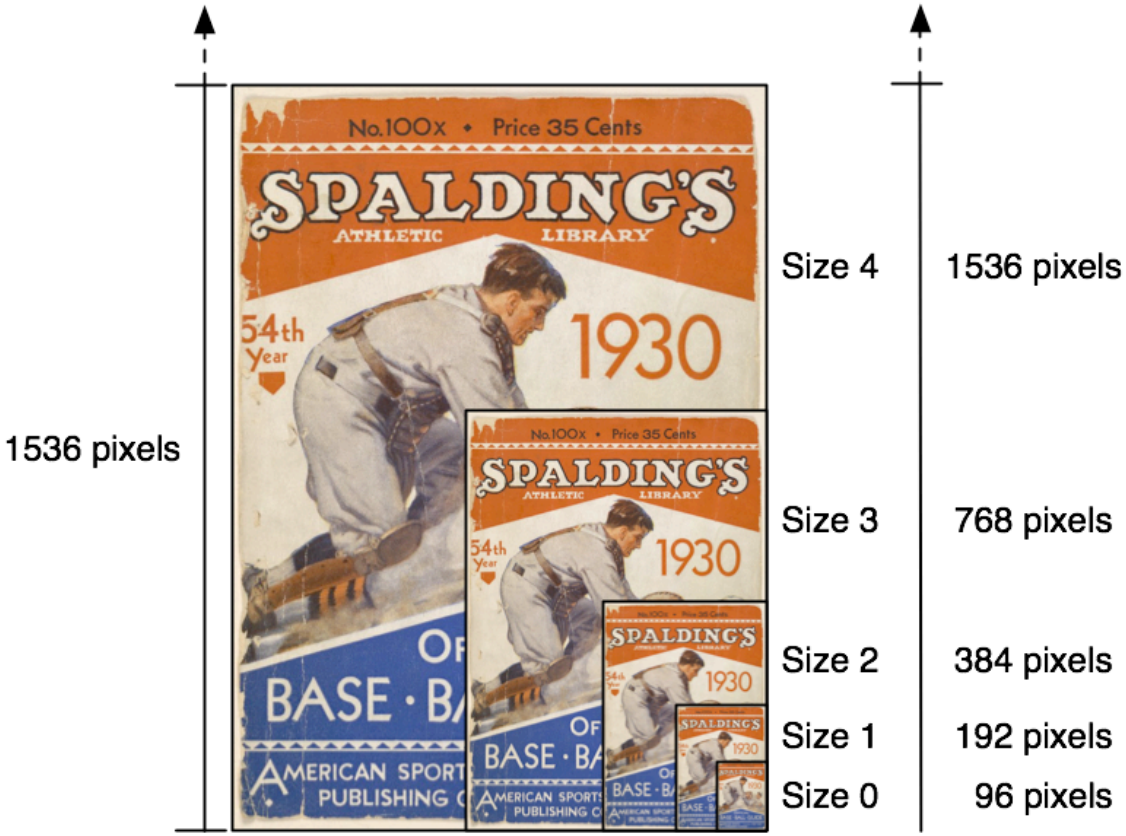
- State-of-the-art compression techniques based on wavelet technology.
- Open Standard Specification
- License-Free: Implementable without payment of royalty and license fees.
- Compression: Mathematically Lossless, Visually Lossless, & Lossy
- Superior compression performance
- Multiple resolution representation
- Random code-stream access and processing
- Rich Metadata Support
- Scalable: Multiple versions can be extracted from a single compressed file.

aDORe djatoka Architecture



Compression: Resolution Levels

- djatoka dynamically determines the number of resolution levels
  - # of times an image can be halved from max(w,h) to 92 pixels or less.
- 92 pixels derived from Kodak PhotoCD Base resolution size.



Compression: Quality

- Utilizes rate-distortion slope threshold values to achieve a specific level of "Image Quality", regardless of subject matter. Also supports absolute rates.
- Number of quality layers and rate-distortion slope threshold values are configurable.

23:1

William-Adolphe Bouguereau

9:1

Baseball Guide (LoC)

8:1

Sargis Ptisak, Gospel of Mark

5:1

Ansel Adams - Manzanar War Relocation (LoC)

Compression: Random Access Efficiencies

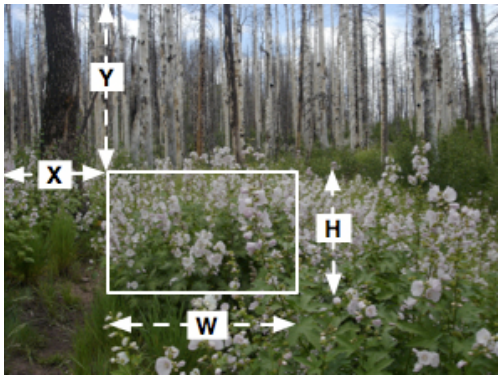
- Uses precinct, instead of tiles, to handle random access efficiencies.
  - Tiles are built into the codestream, while precinct data can be changed without recompressing the image. Both are supported for extraction.
- Packet Length-Tile (PLT) Markers are added to improve extraction times.
- A RPCL (Resolution-Position-Component-Layer) order is applied.

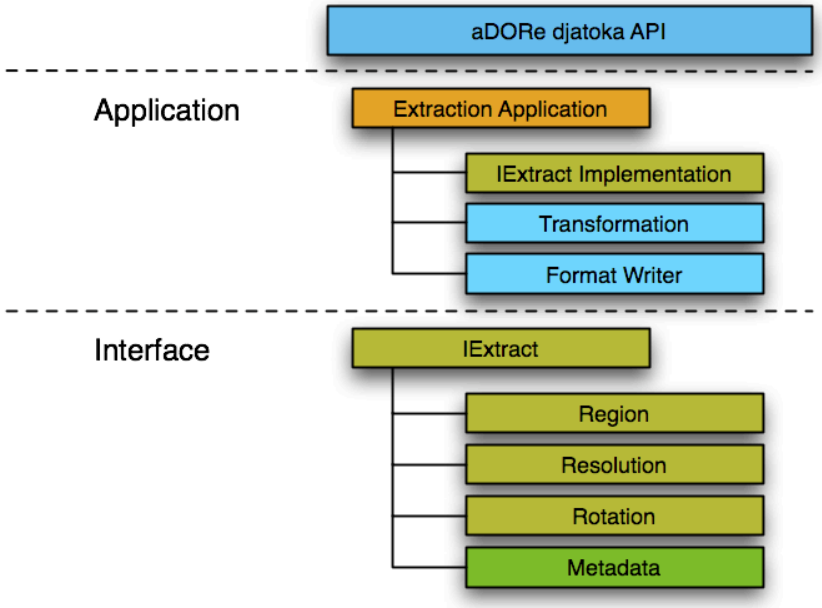
Precinct Structure

Tile Structure

Extraction Features

- Application and API provides the current capabilities:
  - Resolution & Region Extraction
  - Rotation
  - Support for a rich set of input/output formats (e.g. JPG, PNG, TIF, JPEG 2000)
  - Extensible interfaces to perform image transformations (e.g., watermarking)





Why OpenURL?

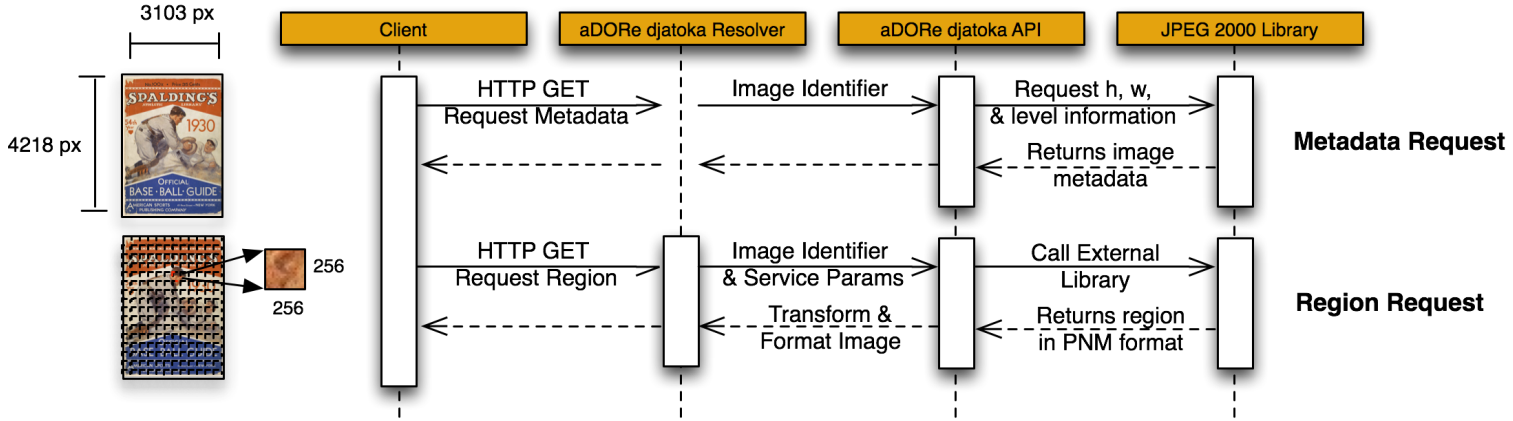
- Existing solutions provide URI-addressability of specified regions, but…
  - Offer limited extensibility for identifier resolution / dissemination services
  - Use home grown HTTP URI syntaxes
- Helpful to have standardized syntax to request Regions or other services.
- Since URIs serve the purpose of requesting services pertaining to an identified resource (the entire JPEG 2000 image), the OpenURL Framework provides a standardized foundation..
- OpenURL provides an easily extensible dissemination service framework.
- Availability and familiarity with OCLC's Java OpenURL package, an open source OpenURL Service Framework.
- Also, to present an alternate Use Case for the OpenURL Framework.

OpenURL Services & Formats

- ContextObject carries information only about a Referent and a ServiceType
  - *info:lanl-repo/svc/getRegion*: the service to request a Region.
  - *info:lanl-repo/svc/getMetadata*: the service to request image metadata.
- JPEG 2000 Region Extraction Service Format
  - Currently registered for Trial Use in the OpenURL Registry

Parameter	*Description*
format	String. Mime type of the image format to be provided as response. Default: image/jpeg
rotate	Integer. Rotates image by 90/180/270 degrees clockwise. Default: 0
level	Integer. Where 0 is the lowest resolution with each increment doubling the image in size. Default: Max level of requested image, based on the number of Discrete Wavelet Transform (DWT) decomposition levels.
region	Format: Y,X,H,W. Y is the down inset value (positive) from 0 on the y axis at the max image resolution. X is the right inset value (positive) from 0 on the x axis at the max image resolution. H is the height of the image provided as response. W is the width of the image provided as response. All values may either be absolute pixel values (e.g. 100,100,256,256), float values (e.g. 0.1,0.1,0.1,0.1), or a combination (e.g. 0.1,0.1,256,256).

aDORe djatoka Sample Service Request



Metadata Request

http://../resolver?url_ver=Z39.88-2004
&rft_id=info:loc-repo/i/00001
&svc_id=info:lanl-repo/svc/getMetadata

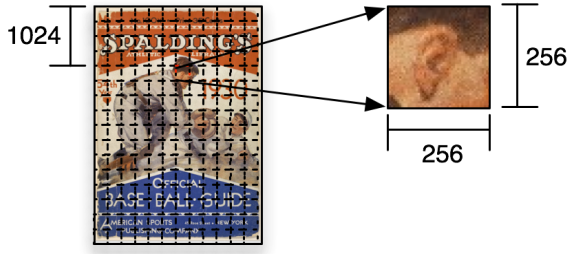
Metadata Response

{
"identifier": "info:loc-repo/i/00001",
"width": "3103",
"height": "4218",
"levels": "6"
}

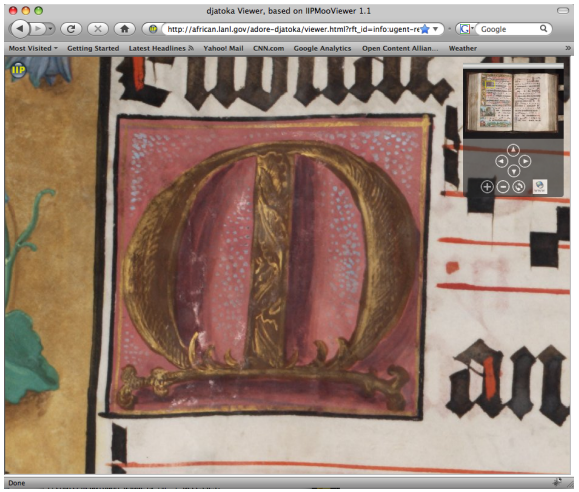
Region Request

http://../resolver?url_ver=Z39.88-2004
&rft_id=info:loc-repo/i/00001
&svc_id=info:lanl-repo/svc/getRegion
&svc_val_fmt=info:ofi/fmt:kev:mtx:jpeg2000
&svc.format=image/jpeg
&svc.level=6
&svc.rotate=0
&svc.region=1024,1536,256,256

Request Response



Client Implementations

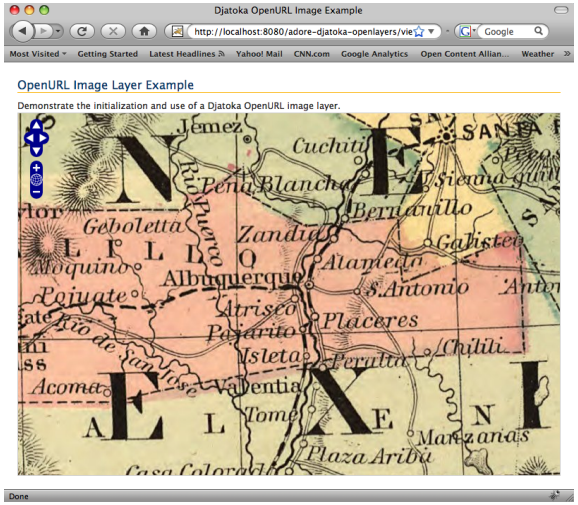


IIP Image djatoka Viewer

- Ajax-based client reference implementation
- Tile-based viewer, similar to Google Maps
- HTML / CSS / Javascript
- Asynchronous djatoka region requests
- Distributed under a GPL Free Software License

OpenLayers djatoka Viewer

- Ajax-based client reference implementation
- Tile-based viewer, similar to Google Maps
- Put an image widget on any web page
- HTML / CSS / Javascript
- Provides OpenURL Support for OpenLayers
- Asynchronous djatoka region requests
- Distributed under a BSD-style License
- Credits to Hugh Cayless (UNC Chapel Hill)



Los Alamos NATIONAL LABORATORY

Where are the resources?

- Referent Resolver
  - For locally managed JPEG 2000 content, the default implementation uses a tab delimited text file to define content identifier to file path mappings.
    - e.g. info:lanl-repo/ds/12345 /smnt/images/12345.jp2
    - Pass in the content identifier as the rft_id and the service will obtain the file handle for the associated image file.
  - For remote image files not under your control, the default implementation can access any resolvable http, ftp, or file URI, download the resource, convert it to JPEG 2000, and store a locally cached version associated with the originally requested URI.
  - New implementations can be easily created to plug djatoka into your existing image database or institutional repository system.

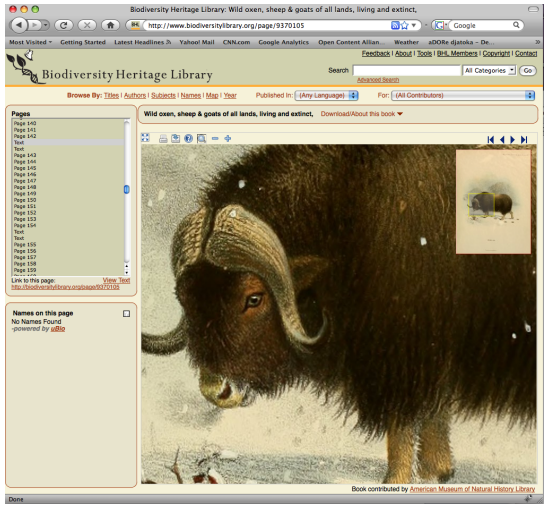
djatoka v1.0 Key Features

- Compression of JPEG 2000 files using properties to improve extraction performance and provide good compression / quality balance.
- Dynamic extraction of multiple resolutions and regions.
- Serialization Plug-in Framework (e.g., BMP, GIF, JPG, JP2, PNG)
- Transformation Plug-in Framework (e.g., watermarking)
- A rich service framework to facilitate the transfer of service parameters via an OpenURL compliant HTTP GET request.
- Configurable File-based Caching for improved performance.

djatoka v1.0 Release Statistics

- Introduced in September 2008, D-Lib Magazine article
- Software also released in September 2008
- Since release:
  - > 400 downloads since release
  - > 450 unique institutions who have visited more than once
  - As of today: 4,838 visits came from 1,282 network locations
- Interest from major cultural heritage and science institutions
- Currently being used in production to serve > 10 million images
- Active efforts to integrate with Fedora and Drupal
- Active efforts to develop additional client implementations (e.g. Flex)

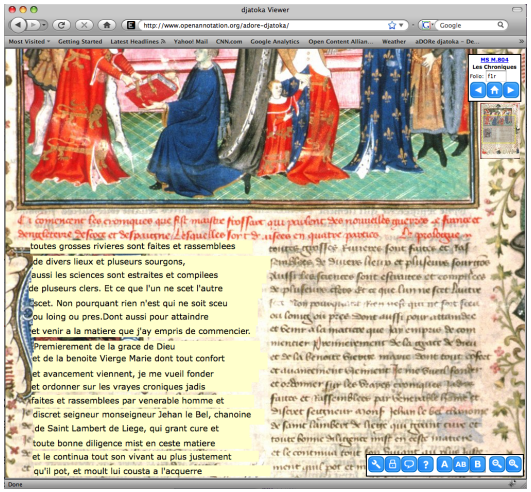
Djatoka at the Biodiversity Heritage Library

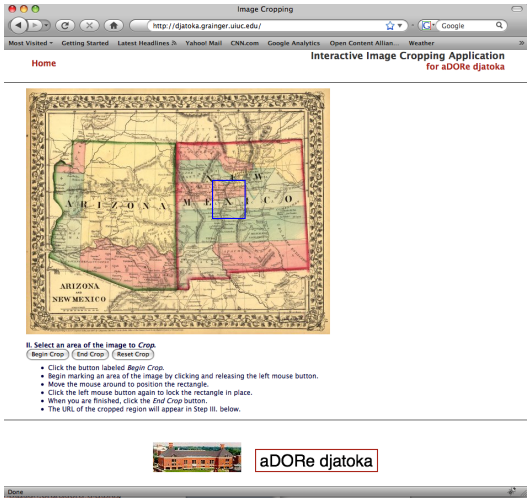


- Running in production since mid-January, 2009.
- Serving nearly 11 million pages.
- Adapted djatoka IIPImage Viewer to fit seamlessly in BHL interface
- Special Thanks to Chris Freeland, Chris Moyers, and Phil Cryer for their support and courage to be such early adopters.
- View the collection at: http://www.biodiversitylibrary.org

- Now serving all page images via djatoka (Freeland, C. & Moyers, C.) http://biodiversitylibrary.blogspot.com/2009/01/now-serving-all-page-images-via-djatoka.html
- HOWTO: serve jpeg2000 images with a scalable infrastructure (Cryer, P.) http://dailyscour.com/blog/howto-serve-jpeg2000-images-scalable-infrastructure

Djatoka and ProjectBamboo





Djatoka-based Manuscript Explorer Demonstrator

- Shows the manuscript pages using Djatoka
- Mouseing over the pages brings up the transcription for the manuscript lines.
- Work of Rob Sanderson (University of Liverpool)
- View demo at: http://www.openannotation.org/adore-djatoka/

Djatoka-based Image Cropping Demonstrator

- Reusing, cropping and referencing digital images
- Demo by Tim Cole (University of Illinois at Urbana-Champaign)
- View demo at: http://djatoka.grainger.uiuc.edu/

djatoka v1.1 Key Features

- JP2 XML Box Support
- Post-extraction Scaling Support
- Added JPX compositing layer extraction support
  - (i.e. access to JPX frames)
- Performance Improvements
- Bug Fixes
- Checks if bitstream is JPEG 2000 format, no ext. necessary.

Current and Future Development

- Online Compression Service
- Embedded Annotation Service
- ICC Color Profile Support
- ORE Serialization Service (Presentation / Application State)
- Repository Integration
  - aDORe
  - Fedora

Technical Requirements

- Sun Java 2 Standard Edition 1.5+
- Tomcat 5.5+
- Ideal:
  - > 512MB RAM
  - Multiple CPUs/cores - Significant Parallel Processing Benefits

Licensing

- Djatoka Image Server and Framework distributed as Open Source under a LGPL License
- Kakadu JPEG 2000 compression / extraction library
  - Free for Non-Commercial use
  - ~8,500 - ~35,000 USD for commercial license.
- Kakadu Binaries provided for:
  - Win32, Mac OS-X x86, Linux x86_32/64, Sparcv9
- Djatoka IIPimage Viewer is a modified IIPMooViewer instance distributed as Open Source under a GPL License.

  http://iipimage.sourceforge.net/
- Djatoka OpenLayers Viewer is a modified OpenLayers build, released under the Clear BSD license.

  http://www.github.com/hcayless/djatoka-openlayers-image-viewer

Demonstrations

Part 2 - JPEG 2000: Barriers to Adoption

JPEG 2000: Barriers to adoption

1. Lack of a clearly recognizable technology champion.
2. Lack of clear guidelines for general and content-specific compressions settings.
3. Lack of an implementation agnostic API for JPEG 2000 compression / extraction.
4. Lack of an open-source service framework, upon which rich WEB 2.0 style apps can be developed.
5. Lack of educational outreach.
6. Legal Concerns

Lack of a clearly recognizable technology champion.

Who is using JPEG 2000?

- Library of Congress
- Biodiversity Heritage Library
- Internet Archive
- Harvard University Library
- National Archive of Japan
- UK National Archives
- British Library
- BBC
- Library and Archives Canada
- Luna Imaging's Insight Installations
- OCLC's ContentDM Installations
- Quite a list, and these are only cultural heritage organizations.
- … but, no one is taking a technology evangelist role.

Lack of guidelines for compressions settings

- "JPEG2000 Implementation at Library and Archives Canada (LAC)"
  - Pierre Desrochers and Brian Thurgood
- LAC JPEG2000 Codestream Parameter Profiles, based on testing:
  - Production/Access Master Profile for Newspapers/Microfilm/Textual
  - Production/Access Master Profile for Color Images/Photographs/ Fine Art/Prints/Drawings/Maps
  - Archival Master Profile for Color Images/Photographs/Fine Art/ Prints/Drawings
  - Archival Master Profile for Cartographic Images

  http://www.archimuse.com/mw2007/papers/desrochers/
- National Digital Newspaper Program (NDNP)
  - JPEG2000 Historic Newspaper Profile

  http://www.loc.gov/ndnp/pdf/NDNP_JP2HistNewsProfile.pdf
- Djatoka Production/Access Master Default Compression Profile
- These are good places to start to develop best practices.

Lack of an implementation agnostic API

- Why is this a barrier?
  - Instead of talking about the format, people tend to talk about the implementations (e.g. Kakadu vs. Aware).
  - A common interface for JPEG 2000 compression and extraction helps ensure format portability and support.
  - Djatoka currently uses Kakadu as the default compression / extraction library, but an interface is provided for alternate implementations (i.e. Aware, OpenJpeg, etc.).
  - Without an abstract interface, new functionality may become dependent on a particular implementation.
- Same reasons exist for lack of an open-source service framework.

JPEG 2000 vs. JPEG vs. PNG vs. TIFF

	JPEG 2000	JPEG	PNG	TIFF
lossless compression performance	+++	+	+++	+++
lossy compression performance	+++++	+++	-	+
progressive bitstreams	+++++	++	+	-
region of interest (ROI) coding	+++	-	-	-
random access	++	-	-	-
low complexity	++	+++++	+++	++
error resilience	+++	++	+	++
genericity	+++	++	+++	++

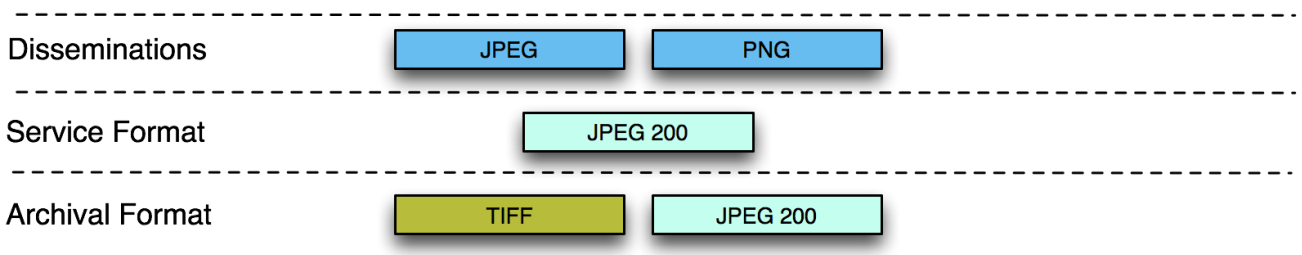
From: http://www.jpeg.org/public/wg1n1816.pdf
& doi:10.1045/july2008-buonora

General Education: Where does JPEG 2000 fall in the file format spectrum?

JPEG 2000 vs. JPEG vs. PNG vs. TIFF

When to use which format?

- JPEG – When lossy compression is of interest and ubiquitous support is the highest priority (e.g. network-based client viewers).
- PNG – When lossless compression is of interest, and content has many pixels of the same color (e.g. vector graphics)
- TIFF – Our security blanket for pixel information, for now.
- JPEG 2000 – When you need a flexible solution, combining good compression and rich dissemination features. Capable of archival role, but more operating system and client application-level support is necessary.



JPEG 2000 Legal Questions

- License-Free
  - From the JPEG committee: “It has always been a strong goal of the JPEG committee that its standards should be implementable in their baseline form **without payment of royalty and license fees**….”
  - Agreements with organizations involved with the standard to allow **use of their intellectual property** in context of the standard.
- Barrier to adoption… **Fear**
  - **“Submarine Patents”**, that some unknown company with patent may come out of blue.
  - Worst case… Embargo format and find solution using TIFF for a few years. Patent terms (20 years in the U.S.) are measured from the original filing.
  - Hasn’t scared Hollywood or the medical industry.

JPEG 2000: Recent Survey

- Digital Project Staff Survey of JPEG 2000 Implementation in Libraries
- David Lowe and Michael J. Bennett, University of Connecticut Libraries

In general the results indicate...

- People, even in the field of digital imaging, don't have a very good understanding of the JPEG 2000 format and its features.
- Why aren't people using JPEG 2000 for their digitization projects?
  - Lack of general education materials focused on cultural heritage use cases.
  - Legal concerns.
  - Lack of JPEG 2000 compression option guidelines.
  - Lack of desktop application support.
  - Lack of open-source & free implementations for compression/extraction.
  - Lack of open-source & free JPEG 2000 image server.
- Supports the need for…
  - Education materials and case studies illustrating the benefits of JPEG 200 for both preservation and access.
  - Prescribed compression setting profiles for different types of content.
  - More open-source JPEG 2000 application support.

Conclusions

- JPEG 2000 has amazing potential as a service format
- Need to invest time and effort into making the format work.
  - Develop working groups to define compression profiles.
  - Develop case studies illustrating benefits of JPEG 2000.
    - JPEG 2000 as a service format
    - JPEG 2000 as a preservation format
    - Reduction in storage costs
    - Simplification of content management
    - Dissemination service options
  - Fund open-source server/client development efforts.
  - Fund and improve open-source compression/extraction libraries

Thank You

- Please feel free to contact us and thank you for your support.
- Available at:
  http://african.lanl.gov/aDORe/projects/djatoka
- SourceForge effort at:
  http://sourceforge.net/projects/djatoka
- Demonstrations at:
  http://african.lanl.gov/adore-djatoka/